RECEIVED
CENTRAL FAX CENTER

APR 2 8 2008

PATENT
APPLICATION 09/822,300
ATTORNEY DOCKET 2000P07515US01 (1009-087)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	:	Muenzel, Georg
Application #	:	09/822,300
Confirmation #	:	8037
Filed	:	23 March 2001
Application Title	:	Industrial Automation System Graphical Programming
		Language Storage and Transmission
Art Unit #	:	2193
Latest Examiner	:	Vu, Tuan A.

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.132

Sir:

I, Georg Muenzel, a citizen of Germany, whose full post office address is 75 Edgemere Ave, Plainsboro, NJ 08536 declare as follows under penalty of perjury.

Background

1. I hold a Diplom (Univ.) degree, which is equivalent to a Master's degree, in Mathematics from University of Erlangen-Nuernberg, Germany, awarded in 1982.

2. I am currently a Program Manager with Siemens Corporate Research, Princeton.

3. Since 1984, I have worked continually in the field of Industrial Automation, with particular emphasis in industrial automation engineering.

4. During my career, I have been granted one U.S. patent for my own inventions in the field of industrial automation systems.

Page 1 of 12

PATENT
APPLICATION 09/822,300
ATTORNEY DOCKET 2000P07515US01 (1009-087)

Review

5. I have reviewed Application Serial No. 09/822,300 (hereinafter the present application).

6. I know, and limit my statements herein to, what one of ordinary skill in the art of the present application would have known on the priority date claimed by the present application (24 March 2000).

7. I have reviewed the USPTO Office Action dated 31 January 2008 (hereinafter the "Office Action") regarding Application Serial No. 09/822,300.

8. I have reviewed U.S. Patent 6,634,008 ("Dole").

9. I have reviewed U.S. Patent 7,089,530 ("Darkinski").

10. Among the subject matter with which I was familiar prior to 24 March 2000 was subject matter of the type recited in Dole.

11. Among the subject matter with which I was familiar prior to 24 March 2000 was subject matter of the type recited in Darkinski.

Dole is Not Pertinent Art to the Claimed Subject Matter of Claim 54

12. The claimed subject matter of claims 54 relates to a method that comprises "automatically converting an industrial automation computer program" that is "adapted for controlling an industrial process via a programmable logic controller" "from a first version written in an industrial automation graphical programming language that is not interpretable by a web browser to a second version written in a web-browser interpretable hierarchical textual markup language."

13. Dole is directed toward (emphasis added), "[a]n **environment for designing integrated circuits.** Computers include browsers for displaying pages of forms, with the computers in communication with a methodology server and a compute server. The methodology server contains design methodologies accessed by the computers, with the design methodologies defining steps of designing and testing of integrated circuits. The computers or methodology server are also in communication with a

Page 2 of 12

compute server. The compute server executes electronic design automation tools as requested." *See*, Abstract.

14. One having ordinary skill in the art would not have found, as of 24 March 2000, "[a]n **environment for designing integrated circuits**" to be reasonably pertinent to any problem related to the claimed subject matter of claim 54, and in particular not to any disclosed problems associated with a method that comprises "automatically converting an industrial automation computer program" that is "adapted for controlling an industrial process via a programmable logic controller" "from a first version written in an industrial automation graphical programming language that is not interpretable by a web browser to a second version written in a web-browser interpretable hierarchical textual markup language."

Dole in View of Dardinski and/or Allegedly Admitted Prior Art Do Not Render Claim 54 Obvious

15. Claim 54 states, *inter alia*, "automatically converting an industrial automation computer program from a first version written in an industrial automation graphical programming language that is not interpretable by a web browser to a second version written in a web-browser compatible hierarchical textual markup language, each of said first version and said second version adapted to be compiled into machine-executable code adapted for controlling an industrial process via a programmable logic controller".

16. Regarding previously pending claim 1 of the present application, the Office Action states, at Pages 8-9, "Dole discloses ... converting the internal representation to a markup language version of the code industrial automation computer program (e.g. HTML, CGI - col. 7, lines 26-42; Fig 10; *more desirable to use XML* -col. 16, lines 10-47; Fig. 13)."

17. One having ordinary skill in the art would have found the statements of the present Office Action recited in paragraph 16 factually incorrect as of 24 March 2000.

Page 3 of 12

18. Instead, in order to construe the phrase "industrial automation control program", one having ordinary skill in the art would have reviewed the present application at least at page 1, lines 24-27, which states, "[g]raphical programming languages are widely used in the field of industrial automation. They provide an intuitive way for automation engineers to specify the control logic for an industrial control application to be run by a controller, usually a programmable logic controller ("PLC")."

19. One having ordinary skill in the art would have equated the "industrial control application to be run by a controller" as recited in paragraph 18 with the "industrial automation computer program" of previously pending claim 1 and presently pending claim 54.

20. Based at least upon the portion of the application recited in paragraph 18, one having ordinary skill in the art would have interpreted the phrase "industrial automation computer program" to mean a computer program that is capable of running on a programmable logic controller that specifies control logic for an industrial control application.

21. One having ordinary skill in the art would have found that col. 7, lines 26-42 of Dole states, "[t]he interface and flow control tool encompasses HTML pages and CGI scripts. The HTML pages include input forms for defining methodologies, including steps of methodologies, as well as chip and block home pages and executable methodologies. The CGI scripts receive and act on data input to the input forms to create files defining methodologies, chips and blocks, and executable methodologies attached to chips and blocks. The CGI scripts also cause execution of electronic design automation (EDA) tools residing on the compute servers (illustrated in FIG. 2). Accordingly, the design server contains files 303. The files are created by the CGI scripts in response to input to the input forms applying new methodologies, and responsive to input to input forms attaching methodologies to chips or blocks. In addition, in one embodiment the files include files and libraries comprising design data formed as the result of the execution of the EDA tools."

Page 4 of 12

22. In view of how one having ordinary skill in the art would construe the phrase
"industrial automation computer program" (as indicated in paragraph 20, *supra*), one
having ordinary skill in the art would not have found the portion of Dole cited in
paragraph 21 to teach anything regarding a "second version" of "an industrial
automation computer program" "written in a web-browser compatible hierarchical
textual markup language" that is "adapted for controlling an industrial process via a
programmable logic controller" as claimed by claim 54.

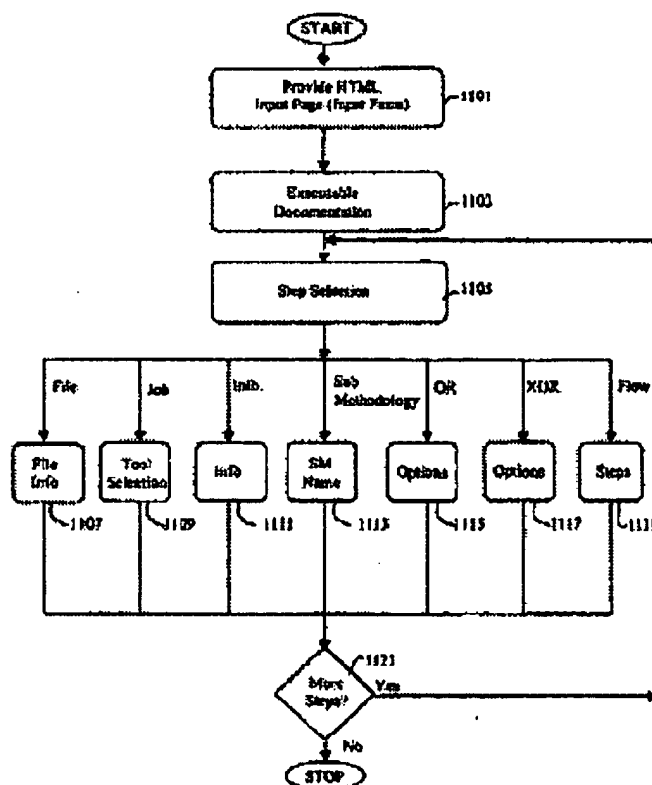23. One having ordinary skill in the art would have found that Fig. 10 of Dole illustrates:

FIG. 10

24. In view of how one having ordinary skill in the art would construe the phrase
"industrial automation computer program" (as indicated in paragraph 20, *supra*), one

Page 5 of 12

having ordinary skill in the art would not have found the portion of Dole cited in paragraph 23 to teach anything regarding a "second version" of "an industrial automation computer program" "written in a web-browser compatible hierarchical textual markup language" that is "adapted for controlling an industrial process via a programmable logic controller" as claimed by claim 54.

25. One having ordinary skill in the art would have found that col. 16, lines 10-67 of Dole states, "[a]lthough HTML based forms are typically used to capture design methodologies, it is often more desirable to use XML (Extensible Markup Language) script to define design methodologies because of advantages that XML has over HTML. In XML, information is divided into useful components called elements, e.g., titles, paragraphs and part numbers. The elements may be formatted, sorted, or searched in consistent fashion. The elements are typically named and defined in a computer program called a Document Type Definition (DTD) Using XML, a methodologist is able to create a single file to describe each design methodology. The single file that describes the design methodology may be used to create other files needed to execute the design methodology. For example, new features can be added to XML over time since XML is an extensible language. In addition, parsers are easy to develop using XML. For example, the parsers may be partially generated automatically from the DTD. Further, XML sources may be scanned by various different programs for different purposes. For example, a source code based on XML may be scanned by a search engine. Another benefit of using XML is that XML is capable of providing multiple language support. For another example, an XML file is easy to create provided that a good DTD has been created. In addition, an XML-based DTD file may be used to specify the internal nature of the XML files used to define design methodologies. Further, XML hyper-linking is more powerful than HTML hyper-linking, and XML hyper-linking may be used to refer to parts of other XML files. Widely used web browsers may not have a capability to display pages having embedded XML. Therefore, in an alternate embodiment, rather than using an

Page 6 of 12

input page to capture design methodology, a methodologist creates an XML script defining a design methodology in a single file. In this embodiment, the XML files are used by Common Gateway Interfaces (CGI's) to drive the integrated circuit design and fabrication system rather than directly viewed using a browser. FIG. 13 is a process of using XML as a design methodology capture script. According to the process in step 521, a methodologist captures a design methodology in a single file in the form of an XML script. Next, a converter with XML parsing capability is used in step 523 to convert the captured design methodology into multiple files including info and index files as well as a directed acyclic graph (DAG) file. As shown in step 525, the methodologist may update the design methodology by modifying the XML file. Thus modified, the XML file may be converted again into multiple files. Use of a single XML file is preferable to using multiple files generated by HTML forms since the single XML is easier to archive and maintain than the multiple files. The process in step 527 may convert the modified design methodology in XML format to the design methodology format where the design methodology comprises multiple files. In an embodiment of the present invention, a converter converts the captured design methodology format having multiple files to a single file having an XML script."

26. In view of how one having ordinary skill in the art would construe the phrase "industrial automation computer program" (as indicated in paragraph 20, *supra*), one having ordinary skill in the art would not have equated a "design methodology" "for designing integrated circuits" with an "industrial automation computer program" as claimed by claim 24.

27. In view of how one having ordinary skill in the art would construe the phrase "industrial automation computer program" (as indicated in paragraph 20, *supra*), one having ordinary skill in the art would not have found the portion of Dole cited in paragraph 25 to teach anything regarding a "second version" of "an industrial automation computer program" "written in a web-browser compatible hierarchical textual markup language" that is "adapted for controlling an industrial process via a

Page 7 of 12

programmable logic controller" as claimed by claim 54.

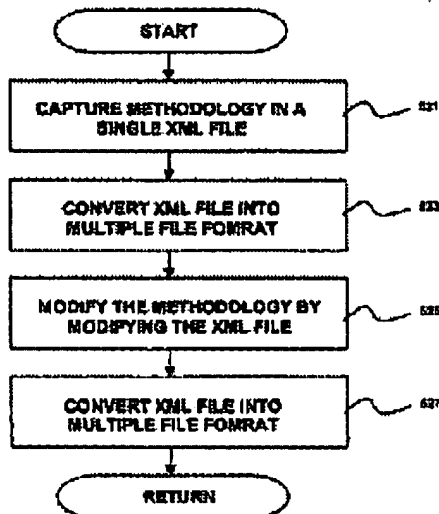28. One having ordinary skill in the art would have found that Fig. 13 of Dole illustrates:

FIG. 13

29. In view of how one having ordinary skill in the art would construe the phrase "industrial automation computer program" (as indicated in paragraph 20, *supra*), one having ordinary skill in the art would not have found the portion of Dole cited in paragraph 28 to teach anything regarding a "second version" of "an industrial automation computer program" "written in a web-browser compatible hierarchical textual markup language" that is "adapted for controlling an industrial process via a programmable logic controller" as claimed by claim 54.

30. One having ordinary skill in the art would have found that col. 16, line 65 – col. 17, line 2 of Dole states, "[i]n an embodiment of the present invention, a converter converts the captured design methodology format having multiple files to a single file having an XML script. A specification of a Methodology Document Type Definition (DTD) is used during development of these converters."

31. In view of how one having ordinary skill in the art would construe the phrase

Page 8 of 12

"industrial automation computer program" (as indicated in paragraph 20, *supra*), one having ordinary skill in the art would not have found the portion of Dole cited in paragraph 30 to teach anything regarding a "second version" of "an industrial automation computer program" "written in a web-browser compatible hierarchical textual markup language" that is "adapted for controlling an industrial process via a programmable logic controller" as claimed by claim 54.

32. The Office Action relies upon allegedly "admitted prior art" from the present application in an attempt to overcome admitted deficiencies of Dole regarding the claimed subject matter of claim 1.

33. Regarding claim 1, the Office Action asserts, at Pages 9-10, "Dole does not explicitly disclose that the industrial automation program is to control a programmable logic controller (PLC). At the time the invention was made, embedding complex system in a single chip - IC - such as endeavored by Dole (see Dole: *system-on-chip* - BACKGROUND) such that complex controlling micro chip devices, e.g. integrated circuits or microcontroller being a field for developers to develop automation control code to test their functions was a known concept. Dole methodology to design complex controller chip wherein design is based on graphical selection regarding chip related functions or architecture (*see place and route* -Fig. 9; tool selection -Fig. 10; *verilog* - Fig. 11; *chip home page* - col. 15, lines 40-62) is thus further evidenced by APA; that is, APA discloses that graphical design by engineers can be supported by graphical programming languages that enable specify control logic of programmable logic controller (PLC - see Specifications , pg, 1). APA is teaching the same line of industrial type of design and/or circuits building/controlling as Dole — that is, in view of the web based and graphic-based tool/methodologies by Dole to. The graphical programming concept is evidenced in Dole's developing of industrial type of design via using graphical tool and methodologies, wherein complex chip(IC) logic or controller functionality can be targeted for being modeled and graphically designed (see Dole: Fig. 28; *Tool 1405* Fig. 5; *Netlist* EDA Tool - Fig. 6; *Flow control tool 315* - Fig. 4 - Note:

Page 9 of 12

graphically assembling of blocks and execution of the control flow of components in a circuit design reads on graphical programming). Based on the fact that one such microcontroller IC type design or complex controller device being such target can be a PLC as set forth by APA from above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to apply circuit synthesis tool, control data communication and web markup conversion as taught by Dole so that the target to be designed would be a control logic of a integrated chip or single controller having control functionality of a PLC such as taught by APA. One would be motivated to do so because the internet based control applied to industrial design and control as endeavor by both Dole using this framework to implement industrial design, programming and testing of a PLC as by APA would enable hardware modeling and design such IC controller as perceived in Dole to put into use the very usefulness of a PLC using the possibilities from a computer technology to implement the graphical programming as set forth above (see APA, pg. 1) and in Dole from above.

34. One having ordinary skill in the art would have found the statements of the present Office Action recited in paragraph 33 factually incorrect as of 24 March 2000.

35. One having ordinary skill in the art would, indeed, have found that "Dole does not explicitly disclose that the industrial automation program is to control a programmable logic controller (PLC)", as admitted by the Office Action.

36. Contrary to the assertions recited in paragraph 33, one having ordinary skill in the art would not have found, for example, that "an industrial control application" that specifies "control logic" to be run by a "programmable logic controller", as stated in the present application, to be in "the same line of industrial type of design and/or circuits building/controlling as Dole — that is, in view of the web based and graphic-based tool/methodologies by Dole to [sic]."

37. Instead, one having ordinary skill in the art would have found Dole to be completely irrelevant to "an industrial control application" that specifies "control logic" to be run by a "programmable logic controller".

Page 10 of 12

38. Moreover, based upon a review of the present application, one having ordinary skill in the art would have recognized that the alleged "graphical tool and methodologies" of Dole "for designing integrated circuits" would have been written in a significantly different language from the "industrial automation computer program" of claim 54.

39. One having ordinary skill in the art would have reviewed Dole at col. 11, lines 62-65 and found that the languages described by Dole for "product specifications" for "designing integrated circuits" were "HDL, VHDL, Verilog".

40. One having ordinary skill in the art would have reviewed pages 1-2 of the present application and found that a "program" written for a "programmable logic controller" are written in languages "such as are defined in IEC 61131".

41. One having ordinary skill in the art would have found that languages defined in IEC 61131 to be significantly different from "HDL", "VHDL", and "Verilog".

42. Because of the differences between languages defined in IEC 61131 and "HDL", "VHDL", and "Verilog", one having ordinary skill in the art would not have been enabled to practice the claimed subject matter of claim 54 based upon the applied portions of the relied-upon references used in rejected each of claims 1-53 by the Office Action.

43. One having ordinary skill in the art would not have found the applied portions of Dardinski, which is relied upon by the Office Action as allegedly teaching "a binary Common Object Model representation converted to obtain the markup language version", overcome the deficiencies of Dole and/or the allegedly admitted prior art regarding the claimed subject matter of claim 54.

44. For at least the reasons indicated in paragraphs 15-43, one having ordinary skill in the art would not have determined that Dole, Dardinski, and/or the allegedly admitted prior art, alone or in combination, teach, "automatically converting an industrial automation computer program from a first version written in an industrial automation graphical programming language that is not interpretable by a web browser to a second version written in a web-browser compatible hierarchical textual markup
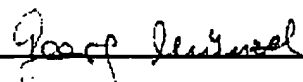
Page 11 of 12

language, each of said first version and said second version adapted to be compiled into machine-executable code adapted for controlling an industrial process via a programmable logic controller" as claimed by claim 54.

45. For at least the reasons indicated in paragraphs 15-43, one having ordinary skill in the art would not have determined that Dole, Dardinski, and/or the allegedly admitted prior art, alone or in combination, enabled the subject matter of, "automatically converting an industrial automation computer program from a first version written in an industrial automation graphical programming language that is not interpretable by a web browser to a second version written in a web-browser compatible hierarchical textual markup language, each of said first version and said second version adapted to be compiled into machine-executable code adapted for controlling an industrial process via a programmable logic controller" as claimed by claim 54.

I further declare that all statements made herein of my own knowledge are true and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code and that willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signed this _25ᵗʰ_ day of ___April___ 2008   _____
                                                          \'Georg Muenzel